GQS: Graph Query System

Thomas Fannes & Jan Ramon

[firstname.lastname]@cs.kuleuven.be

Starting Grant
240186 MiGraNT

European Research Council

Declaratieve Talen en Artificiële Intelligentie

KU LEUVEN

GQS

- Overview
- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance
- Extras & current status

GQS

- Overview

- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance

- Extras & current status

GQS overview

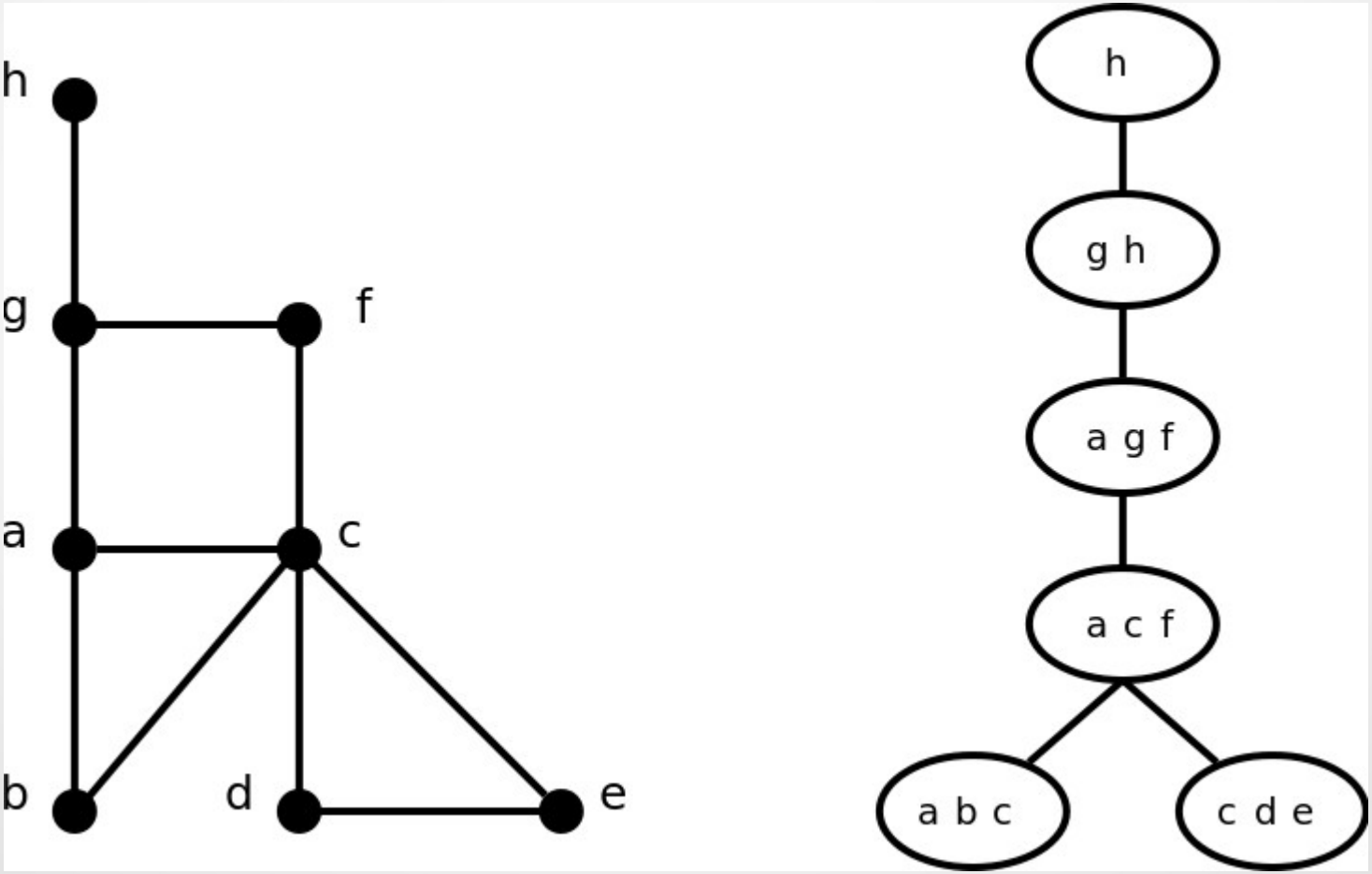
- Many different flavors of graph databases:
  - Single network vs multiple networks
  - Standard graph vs hypergraph
  - Query type:
    - Pattern
    - Embedding
    - Aggregated value
    - …
  - Embeddings under [iso|homo]-morphism

GQS overview

- Why a new graph query system?
- Use state-of-the-art graph algorithms
  - More efficient mining (special settings)
  - Results applicable for data mining
- GQS settings:
  - Single, large network database
  - Rooted, bounded treewidth pattern graphs

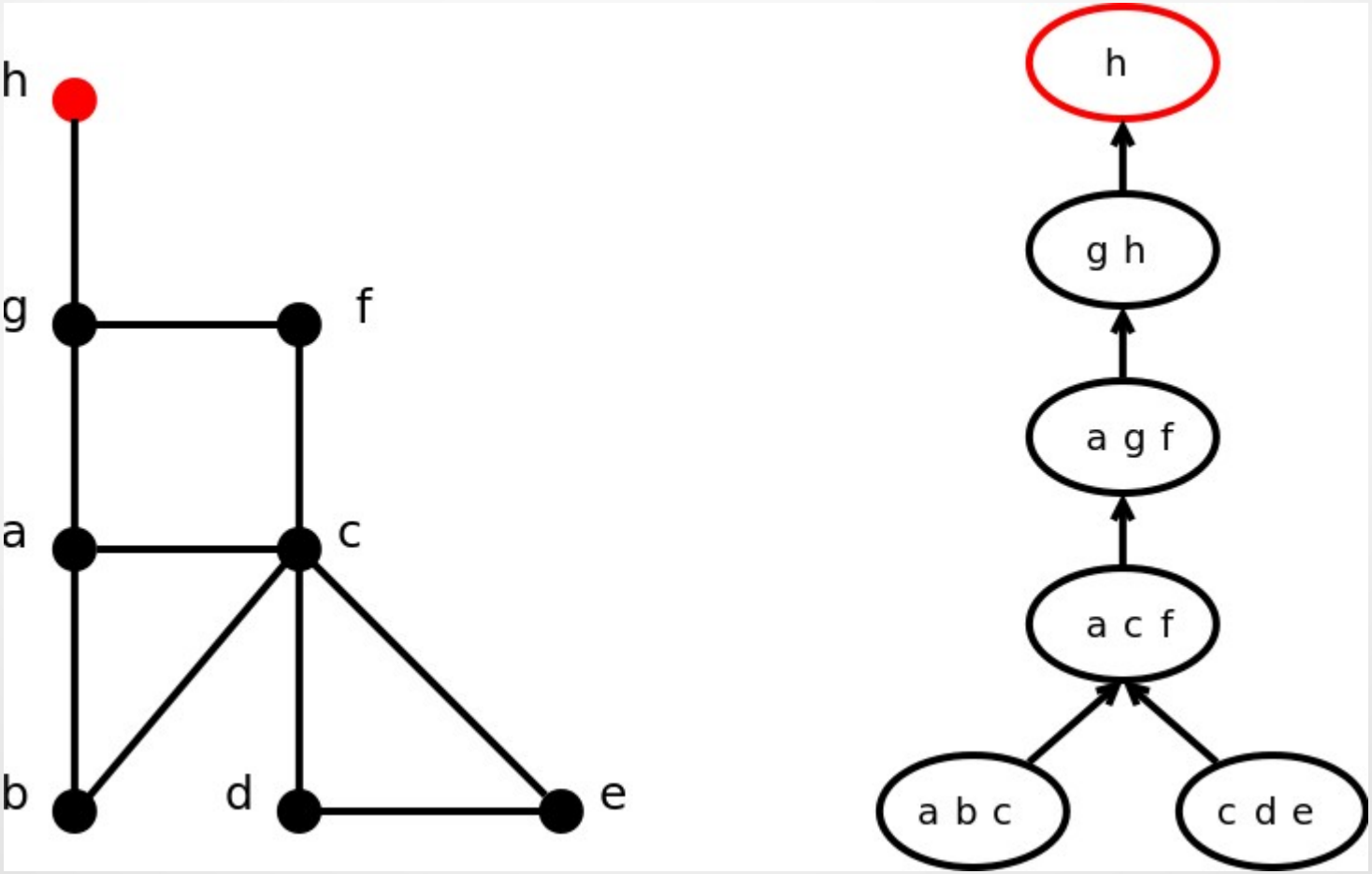
GQS overview

- Rooted bounded treewidth graph



GQS overview

- Rooted bounded treewidth graph



GQS overview

- Standard query operators:
  - List vertex
  - Extend vertex
  - Join
  - Project
  - Select

- Per embedding, calculate a target value:

  Product of values associated with network vertices

GQS overview

- Target value (per embedding):

  Product of values associated with network vertices

- List: $T(\ L(v)\ ) = T(v)$
- Extend: $T(\ E(e,v)\ ) = T(E)*T(v)$
- Join: $T(\ J(e_1, e_2)\ ) = T(e_1)*T(\ e_2 \backslash e_1)$
- Project: $T(\ P(e_1 \cup e_2, e_1)\ ) = \Sigma\ T(e_1)$

GQS

- Overview
- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance
- Extras & current status

GQS: default

Task: List **homomorphic** root embeddings of a pattern

evaluation of a projection of a conjunctive query

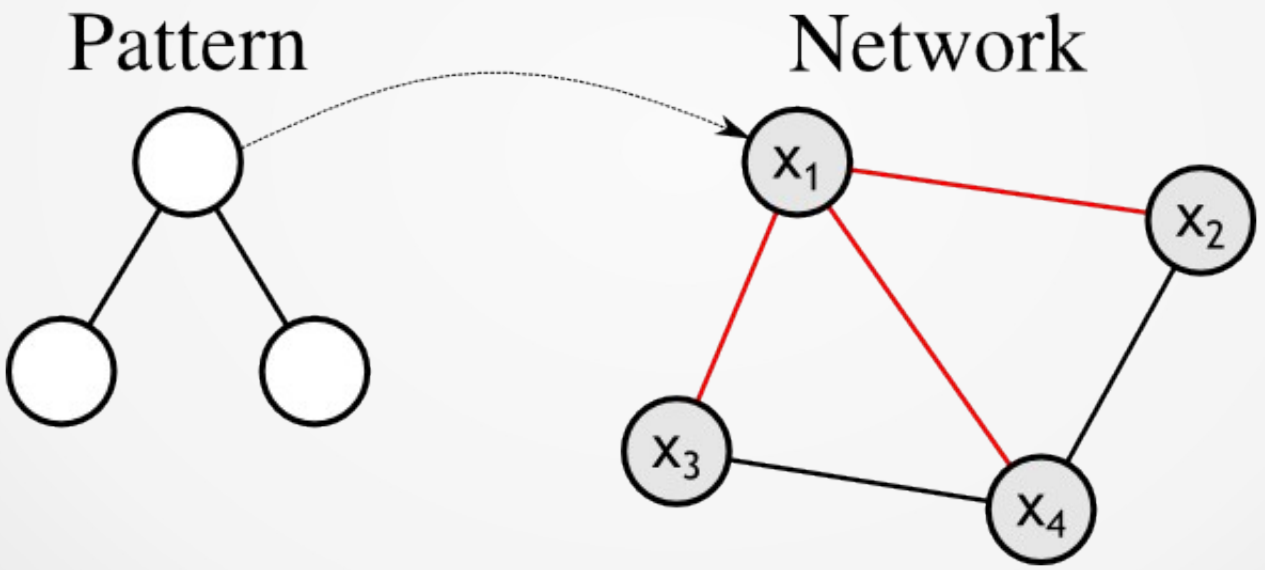
- No target value (or count embeddings per root)
- Runtime:
  - Polynomial in network size
  - Polynomial in pattern size
  - Exponential in pattern treewidth

GQS

- Overview
- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance
- Extras & current status

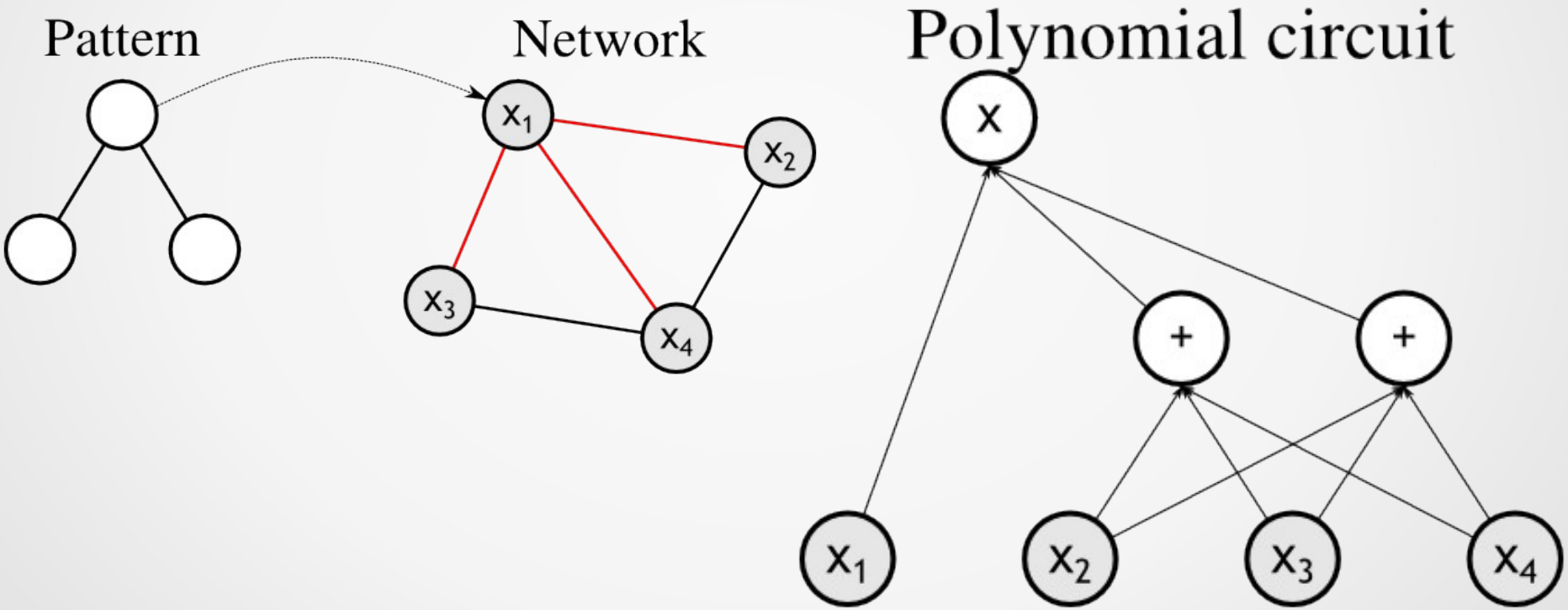
GQS: GF_2

Task: List **isomorphic** root embeddings of a pattern



GQS: GF_2

Task: List **isomorphic** root embeddings of a pattern



$$x_1 x_2^2 + x_1 x_2 x_3 + x_1 x_2 x_4 +$$
$$x_1 x_3 x_2 + x_1 x_3^2 + x_1 x_3 x_4 + x_1 x_4 x_2 + x_1 x_4 x_3 + x_1 x_4^2$$

GQS: GF_2

Task: List **isomorphic** root embeddings of a pattern

$$x_1 x_2^2 + x_1 x_2 x_3 + x_1 x_2 x_4 +$$
$$x_1 x_3 x_2 + x_1 x_3^2 + x_1 x_3 x_4 + x_1 x_4 x_2 + x_1 x_4 x_3 + x_1 x_4^2$$

- Use $GF(2^l)\mathbb{Z}_2^k$ as target value algebra:
  - One embedding $\leftrightarrow$ one term
  - Squares or higher are evaluated to zero
  - Randomized approach:
    - T(e) ≠ 0 → isomorphic embedding of pattern
    - T(e) = 0 → Pr[no isomorphic embedding of pattern] < ε

GQS: GF_2

Task: List **isomorphic** root embeddings of a pattern

- Target value in randomized $GF(2^l)\mathbb{Z}_2^k$ algebra
- Runtime:
  - Polynomial in network size
  - Mildly exponential in pattern size $O(2^{|V(P)|})$
  - Exponential in pattern treewidth

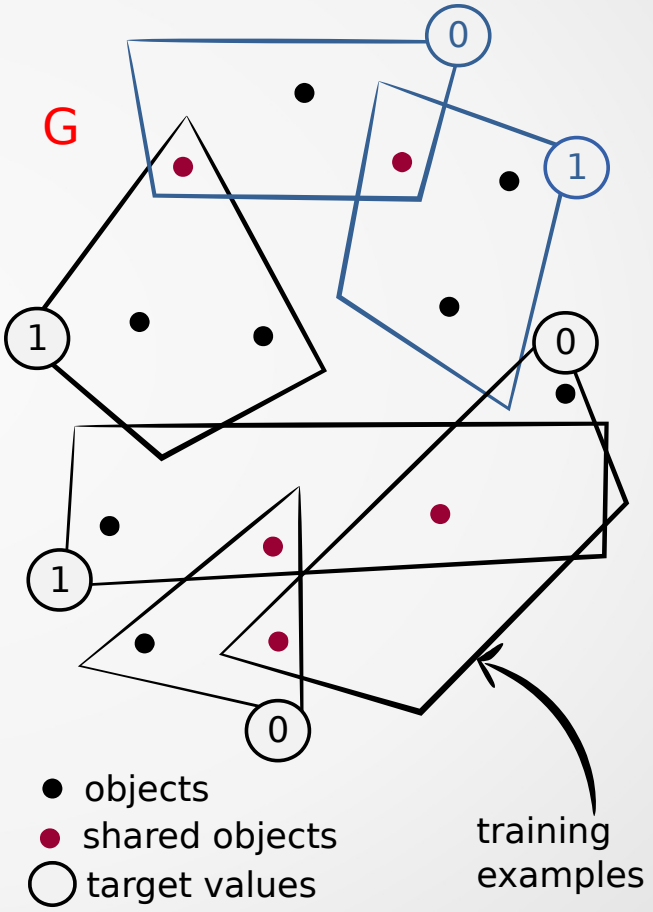
- (Kibriya & Ramon, DMKD 2013)

GQS

- Overview
- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance
- Extras & current status

GQS: s-measure

- Vertices: objects
- Hyperedges: examples
- Hyperedge overlap on shared objects.
  - these examples are not independent
- Independence assumption
  - Hyperedges are fixed.
  - Choose vertex features i.i.d.



GQS: s-measure

- Measuring effective sample size

  Given a networked training set, can we get information out of it equivalent to *n* i.i.d. examples? What is *n*?

- For a pattern P, the s-measure gives a anti-monotonic support measure

- (Wang & Ramon, DMKD 2013)

GQS: s-measure

- Influence of each factor is at most 1:
- max s

  $s = w_1 + w_2 + w_3 + w_4 + w_5 + w_6$

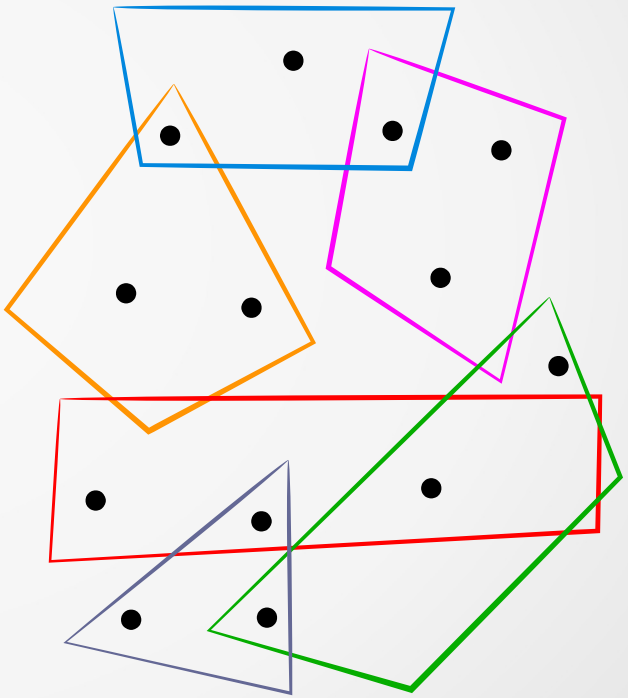
  Subject to

  $w_1 + w_2 \leq 1,\ w_1 + w_3 \leq 1,$

  $w_4 + w_5 \leq 1,\ w_5 + w_6 \leq 1,$

  $w_4 + w_6 \leq 1$

  $w_1, w_2, w_3, w_4, w_5, w_6 \geq 0$



$s = 3.5$

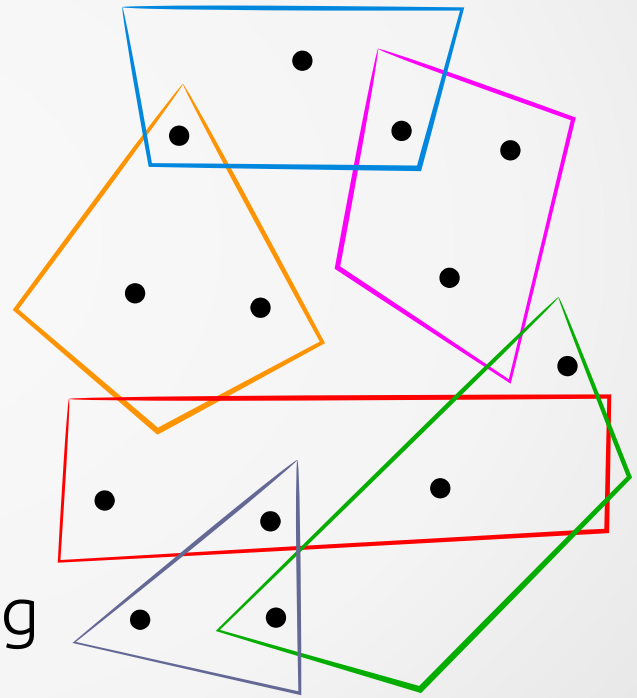
GQS: s-measure

- Linear program → efficient
- Target value?
  - Writes out program!

List root embeddings

Support measure for each pattern

Statistical significance of embedding



s = 3.5

GQS

- Overview
- Target value algebras
  - Homomorphism
  - Isomorphism
  - Support measure & embedding significance
- Extras & current status

GQS: extras

- Different query plan optimizations
  - Memory footprint reducing
  - Optimizations specific for target value algebra

- 2-step approach:

C++ runtime polymorphic system

C++ template-based query program

GQS: Current status

- [x] Query system
- [x] Query optimization

- [x] GF2 isomorphism integration
- [ ] GF2 extended tests

- [ ] s-measure integration
- [ ] s-measure extended tests

GQS

Thanks !

Any questions ?